Alternately, or in addition, the weight may be based on the combined utilization of the candidate diagnosis among all passing system tests. Other factors that may be utilized instead of, or in addition to, those mentioned above, such as an observed variability of the results of passing and failing system tests, may form the basis of the weight of the candidate diagnosis as well.

Generally speaking, the use of such factors in weighting a candidate diagnosis produces a more accurate indication of whether the associated diagnosis is correct. As a result, circumstances in which closely valued or identical weights for multiple candidate diagnoses result are significantly reduced.

Other aspects and advantages of the invention will become apparent from the following detailed description, taken in conjunction with the accompanying drawings, illustrating by way of example the principles of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a high-level flow chart of a method of diagnosing dynamic system faults according to an embodiment of the invention.

FIG. 2 is a flow chart of a method according to embodiments of the invention of generating and weighting candidate diagnoses possibly responsible for an observed test failure.

FIG. 3 is a block diagram representation of the system under test and its constituent parts.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

A high-level description of a method 100 according to an embodiment of the invention for diagnosing complex faults of a system under test (SUT) is shown in FIG. 1. In general terms, given a model of the SUT, a model for each passing and failing test performed on the SUT, and the results of those tests, at least one candidate

As an example, and with specific reference to FIGURE 3 of the drawings, assume failing tests T_1 and T_2, and passing tests T_3, T_4 and T_5, as shown in the following simplified SUT 300 and test models:

Components:

C_1 301: Subcomponents: None

C_2 302: Subcomponents: sc_2 303, sc_3 304, sc_4 305

Shared Functions:

sf1 = {(C_1 301, 0.6)}

sf2 = {(C_2 302::sc_2 303, 0.4)}

sf3 = {(C_2 302::sc_3 303, 0.3)}

sf4 = {(C_2 302::sc_3 303, 0.6)}

Tests:

T_1 = {sf1, sf3}; Result = Fail

T_2 = {sf1, sf2, sf4, (C_2 302::sc_4 305, 0.7)}; Result = Fail

T_3 = {sf1, sf4, (C_2 302::sc_2 303, 0.4), (C_2 302::sc_3 304, 0.5), (C_2::sc_4, 0.2)}; Result = Pass

T_4 = {sf3, (C_1 301, 0.9)}; Result = Pass

T_5 = {sf1, sf2}; Result = Pass

In this case, the coverage element conflict sets for failing tests T_1 and T_2 are CECS(T_1) = {sf1, sf3} and CECS(T_2) = {sf1, sf2, sf4, (C_2::sc_4, 0.7)}.

Continuing with this example, a "structural element conflict set" (SECS) is also associated with each failing test. A structural element conflict set is set of structural elements wherein each element has some coverage, by way of a structural element coverage or a shared function, in the coverage element conflict set for a particular failing test. In this case, the structural element conflict sets for failing tests